AMENDMENTS TO THE SPECIFICATION

Please amend the specification as indicated hereafter. It is believed that the following amendments and additions add no new matter to the present application.

Please replace the paragraph starting on p. 2, line 28 with the following amended paragraph:

[[.]] Briefly described, in architecture, one embodiment of the system, among others, can be implemented as follows. Memory is provided for storing a text format template. A controller is provided. This controller is configured to generate data representing a graphical layout of a data path macro cell. It is also configured to generate a text format template comprising a variable, based upon the data representing a graphical layout of the data path macro cell. [[.]] Further, the controller is configured to substitute a predetermined value for the variable to generate a text format file representative of a data path macro cell having predetermined characteristics. The controller is also configured to generate a data file representing a graphical layout of a data path macro cell represented by said text format file.

Please replace the paragraph starting on p. 5, line 16 with the following amended paragraph:

Fig. 2A is a flow chart illustrating an example embodiment of the method ~~200~~ of the present invention for generating a text format template representative of a macro cell. In this method, a data path macro is defined (202). More particularly, the various characteristics and functionality of the data path macro cell will be determined by the IC designer.

Please replace the paragraph starting on p. 5, line 21 with the following amended paragraph:

Data (graphical data) representation <u>of</u> a graphical layout of the defined data path macro cell is generated (204). This graphical data may be generated by using a macro cell generator program. This macro cell generator program may be, for example, a program written using a programming language, such as C, C++, PERL, RTL or the like. It will, however, be recognized that other programming languages may also be used to accomplish the purposes of the macro cell generator program. In

creating the data path macro cell generator program, the IC designer will incorporate data that specifies such things as the signal names and documentation layer information. Text format data representative of the data path macro cell is generated based upon the graphical data representation (206). This text format data may also be referred to as an "archive" file and may be generated by using one of many commonly available archive tools. This text format data describes the attributes or characteristics of the IC design represented by the graphical data. This text format data will specify values for the various characteristics of the IC represented by the graphical data. These values are constants that are specific to describing the characteristics of the IC layout represented by the graphical data.

Please replace the paragraph starting on p. 6, line 22 with the following amended paragraph:

FIG. 2B is a flow chart illustrating an example embodiment of the method ~~200~~ of the present invention for generating a macro cell based upon a text format template representative of a macro cell. In this method, a text format template is retrieved or read out of memory ~~(214)~~ (212) and the variables set out in the text format template are changed to specify values that correspond to a macro cell having specific characteristics (214) to generate a text format file representative of a macro cell having predetermined/desired characteristics. For example, values may be entered into the text format template to specify a macro cell having a particular macro cell or data bus width as may be desired.

Please replace the paragraph starting on p. 7, line 20 with the following amended paragraph:

FIG. 3A illustrates a text format file 300 generated based upon data representing a graphical layout of a macro cell having certain characteristics specified by constant values therein. This illustration shows only relevant parts of a sample text format file. It can be seen that this text format file contains a number of constant values including constant values 302, 304, 306, 308, 310 and 312. Constant values 302 and 304 are equal to [["4"]] "8", while constant values 308 and 310 are equal to [["8"]] "4". Similarly, constant value 306 specifies the values "2, 3, 5, 6, 7". The constant value 312 specifies the value "8*S1[0]". FIG. 3B illustrates a text format

template that has been created based upon the text format file of FIG. 3A. This illustration shows only relevant parts of a sample text format template. It can be seen that the constant values 302, 304, 306, 308, 310 and 312 of Fig. 3A have been replaced with variables 352, 354, 356, 358, 360 and 362, respectively, in the text template of Fig. 3B. For example, it can be seen that the constant value 302 ~~("4")~~ ("8") has been replaced by a variable 352. Variable 352 ~~specifies " _width_". Similarly,~~ specifies "_*width*_". Similarly, constant value 312 has been replaced by a variable 362 that specifies "_s1place_". This text format template may be generated merely by editing the text format file for a given macro cell to replace constant values with variables. Subsequently, these variables may be changed again to values that will correspond to a macro cell having particular ~~characteristic~~ characteristics that will be specified by the particular values that are used to replace the variables. It will be recognized that the text format file illustrated by FIG. 3A, as well as the text template illustrated in FIG. 3B may constitute, for example, a Verilog format file and may otherwise be created via, for example, use of the Verilog hardware description language.